COMPUTATIONAL GALOIS THEORY: INVARIANTS AND COMPUTATIONS OVER $\mathbb{Q}$

CLAUS FIEKER AND JÜRGEN KLÜNERS

Abstract. Computational Galois theory, in particular the problem of computing the Galois group of a given polynomial is a very old problem. Currently, the best algorithmic solution is Stauduhar's method. Computationally, one of the key challenges in the application of Stauduhar's method is to find, for a given pair of groups $H < G$ a G-relative H-invariant, that is a multivariate polynomial F that is H-invariant, but not G-invariant. While generic, theoretical methods are known to find such F, in general they yield impractical answers. We give a general method for computing invariants of large degree which improves on previous known methods, as well as various special invariants that are derived from the structure of the groups. We then apply our new invariants to the task of computing the Galois groups of polynomials over the rational numbers, resulting in the first practical degree independent algorithm.

1. Introduction

Computational Galois theory, in particular the problem of finding the Galois group of a given polynomial is a very old problem. While various algorithms have been published, so far they are either impractical for groups of size > 1000 due to the requirement of exact representation of an algebraic splitting field, or they are degree dependent. Examples of algorithms of the first kind include for example the naive approach of constructing a splitting field by repeated factorisation as well as more sophisticated methods [21]. Algorithms of the second kind fall broadly into two approaches: a classical approach that aims to characterise the Galois group as an abstract group by building a decision tree using certain indicators (resolvent polynomials) [3, Chapter 6.3] and a newer approach, by Stauduhar [20] where the Galois group is constructed explicitly as a group of permutations of the roots of the polynomial in question. Stauduhar's method roughly works by traversing the lattice of (transitive) subgroups of the full symmetric group from the top ($\mathbf{S}_n$) down to the Galois group of the polynomial. At each step, this is

done through the help of invariants and the high precision evaluation of those.

This paper naturally splits into two parts: the first discussing the problem of finding a useful invariant for each pair of groups, and the second part explaining how this is used to compute Galois groups of polynomials over $\mathbb{Q}$.

Primitive invariants for permutation groups, i.e. multivariate polynomials with a given stabiliser, are among the most important objects in computational Galois theory, they are the central ingredient in Stauduhar's method [9, 10] for the determination of the Galois group of a polynomial f: Given two groups $H < G$ a (G-relative) H-invariant is used to decide if $\mathrm{Gal}(f) \le H^g$ for some $g \in G$ under the assumption that $\mathrm{Gal}(f) \le G$. Furthermore, applications, such as the explicit realization of Galois groups by explicitly computing defining equations for subfields of the splitting field for f rely on invariants as well [15].

While there are a few methods known for the computation of such invariants in the literature, in applications, invariants were mostly the result of ad-hock methods. Generic algorithms, eg. [1, 11] for individual invariants or [13] for the computation of the entire ring of invariants become rapidly unpractical for larger degree permutation groups.

It should be stressed that while invariant theory gives explicit invariants for all pairs of groups $H < G$, the generic results tend to be impractical as the resulting invariants are computationally far too complex.

In what follows, we will give a new, space-efficient algorithm to compute all invariants of a given degree for arbitrary pairs of groups, and for maximal subgroups of transitive groups we give several constructions that allow the determination of efficient invariants in many cases. We then demonstrate that knowledge of the subgroup structure can also be used to find efficient invariants as frequently invariants for some subgroups can be combined to give invariants for others.

Finally, we demonstrate the efficiency and the limits of our methods by considering several examples.

2. Notation

Transitive groups of degree < 32 are denoted by nTm where m is the degree and n is the number of the group in the classification [4] used by both Magma and Gap. For the rest of the article, we fix some positive integer n. The symmetric group on n elements, $\mathbf{S}_n$ acts on the polynomial ring $\mathbb{Z}[\underline{X}] = \mathbb{Z}[X_1, \dots, X_n]$ in n variables via

$$X_i \mapsto X_{\sigma(i)}$$

for $\sigma \in \mathbf{S}_n$ we usually write F^σ for the image under this map. A polynomial $F \in \mathbb{Z}[\underline{X}]$ is called an H-invariant (for some group $H \leq \mathbf{S}_n$) if $F^\sigma = F$ for all $\sigma \in H$. Given two subgroups $H < G \leq \mathbf{S}_n$, we call a polynomial $F \in \mathbb{Z}[\underline{X}]$ a G-relative H-invariant, if its stabiliser $\operatorname{Stab}_G F := \{\sigma \in G \mid F^\sigma = F\}$ in G equals H. A polynomial $F \in \mathbb{Z}[\underline{X}]$ is called an absolute H-invariant if $\operatorname{Stab}_{\mathbf{S}_n} F = H$.

For any subgroup $H \leq \mathbf{S}_n$ we can consider the ring $\mathbb{Z}[\underline{X}]^H$ of absolute H-invariants and also the invariant field $\mathbb{Q}(\underline{X})^H$ of rational functions that are invariant under H.

Remark 2.1. *If $H < G \leq \mathbf{S}_n$ is a pair of subgroups and if $F \in \mathbb{Z}[\underline{X}]$ is a G-relative H-invariant, then*

(1) *As an extension of fields, $\mathbb{Q}(\underline{X})^H$ is a finite extension of $\mathbb{Q}(\underline{X})^G$ of degree*
$$(\mathbb{Q}(\underline{X})^H : \mathbb{Q}(\underline{X})^G) = (G : H)$$
(2) *Furthermore*
$$\mathbb{Q}(\underline{X})^H = \mathbb{Q}(\underline{X})^G[F]$$
that is, F is a primitive element for the extension.

(3) *From the main theorem on symmetric functions it follows that*
$$\mathbb{Z}[\underline{X}]^{\mathbf{S}_n} = \mathbb{Z}[\sigma_1, \dots, \sigma_n]$$
where $\sigma_i = \sum_{1 \leq j_1 < \cdots < j_i \leq n} \prod_{l=1}^{i} X_{j_l}$ are the elementary symmetric functions.

3. Stauduhar's method

In this section we recall the necessary tools from Stauduhar's method. We do this in a slightly more general context which has the advantage that we can combine the information obtained by the resolvent method and by Stauduhar's method.

Let us assume that we are given a monic polynomial $f \in \mathbb{Z}[X]$ of degree n and we would like to compute the Galois group of f. Certainly, the Galois group is a subgroup of $\mathbf{S}_n$ and therefore we can assume that we know a subgroup $G \leq \mathbf{S}_n$ with $\operatorname{Gal}(f) \leq G$. Assume furthermore that we have a proper subgroup $H < G$ and let $F \in \mathbb{Z}[X_1, \dots, X_n]$ be a G-relative H-invariant polynomial. In the following we denote by $G//H$ a set of representatives of right cosets $H\sigma$ of G/H. The following is proved in [20].

Lemma 3.1. *Let F be G-relative H-invariant and assume that* $\mathrm{Gal}(f) \leq G$*, where* $\mathrm{Gal}(f)$ *acts on the roots* $\alpha_1, \dots, \alpha_n$ *in some fixed closure. Then*

$$R_F := \prod_{\sigma \in G//H} (T - F^\sigma(\alpha_1, \dots, \alpha_n)) \in \mathbb{Z}[T].$$

R_F is called the relative resolvent polynomial (corresponding to $H < G$ and F).

Proof. Since $F^H = F$ we see that R_F does not depend on the choice of coset representatives. The polynomial R_F is invariant under G and since $\mathrm{Gal}(f) \leq G$ it is invariant under $\mathrm{Gal}(f)$. Therefore all coefficients of R_F are in $\mathbb{Q}$ and also algebraic integers, thus in $\mathbb{Z}$. □

Suppose that R_F is squarefree and we know a non-trivial factor of R_F in $\mathbb{Z}[T]$. In this situation we can show that the Galois group of f is contained in a proper subgroup of G and therefore we make progress. In case R_F is not squarefree, we apply a Tschirnhausen transformation $t \in \mathbb{Z}[x]$ and compute a new polynomial

$$R_{F,t} := \prod_{\sigma \in G//H} (T - F^\sigma(t(\alpha_1), \dots, t(\alpha_n))).$$

It can be shown that there exist suitable transformations t such that $R_{F,t}$ is squarefree. Furthermore, introducing t amounts to a change of f that will not affect the Galois group.

The case of linear factors in the next theorem is proved in [20], in fact it formed the key technique in the original paper. The possible use of quadratic factors is mentioned on the last page of [19] but is rejected there since the practical group theory would be too complicated. The general statement is also proven in [9, Satz 2.4], although only the case of linear factors is used to determine groups. Higher degree factors are only considered in a verification step.

Theorem 3.2. *In the above situation assume that R_F is squarefree and that we know a factor $A \in \mathbb{Z}[T]$ dividing R_F of degree* $\deg A = m$*. Denote by $\rho : G \to \mathbf{S}_{G/H}$ the permutation action on right cosets G/H. Then there exist $\sigma_1, \dots, \sigma_m \in G$ such that*

$$A(T) = \prod_{i=1}^{m} (T - F^{\sigma_i}(\alpha_1, \dots, \alpha_n)).$$

Denote by B the set of right cosets $\{H\sigma_i \mid 1 \leq i \leq m\}$.

Then $\mathrm{Gal}(f) \leq \rho^{-1}\left(\mathrm{Stab}_{\rho(G)}(B)\right)$.

Proof. $\sigma_1, \ldots, \sigma_m$ are in pairwise different right cosets of G/H since otherwise $F^{\sigma_i} = F^{\sigma_j}$ and the polynomial A is not squarefree. Extend the σ_i to a complete system of representatives $\sigma_1, \ldots, \sigma_r$ of G/H, where $r = (G : H)$. Now let $\tau \in \mathrm{Gal}(f) \le G$ be an arbitrary element. The elements $\tau\sigma_1, \ldots, \tau\sigma_r$ are also a set of representatives of G/H. Since A is invariant under $\tau \in \mathrm{Gal}(f)$ we get that for $1 \le i \le m : \tau\sigma_i \in H\sigma_j$ with $j \le m$. Therefore we get that $\rho(\tau) \in \mathrm{Stab}_{\rho(G)}(B)$. □

We can also apply this theorem if we know more than one factor.

Corollary 3.3. *Assume that our squarefree R_F factorises via $R_F = A_1 \cdots A_s \in \mathbb{Z}[T]$. Denote by B_i the set of right cosets of G/H corresponding to A_i. Then*

$$\mathrm{Gal}(f) \le \bigcap_{i=1}^{s} \rho^{-1}(\mathrm{Stab}_{\rho(G)} B_i).$$

A special case occurs, if we find a single linear factor. In this case we can describe the stabiliser in more detail.

Corollary 3.4. *Assume that R_F is squarefree and has a linear factor in $\mathbb{Z}[T]$ corresponding to $H\sigma$ for $\sigma \in G$. Then $\mathrm{Gal}(f) \le H^\sigma := \sigma^{-1}H\sigma$.*

Proof. Note that a point stabiliser of $\rho(G)$ is isomorphic to H. □

We remark that in the following we mostly use this Corollary since finding linear factors is much easier than doing a complete factorisation. In particular, we are frequently able to find linear factors without ever constructing R_F completely.

Nevertheless, there are situations where the ability to work with factors of degree > 1 is desirable. Mainly, this happens if we want to work with maximal subgroups $U < G$, where the index $(G : U)$ is huge. If we apply Stauduhar's method to such a group pair, then the degree of the corresponding R_F is $(G : U)$. Furthermore coefficient bounds that we need in our algorithm are dependent on this index. In such a case it could be nice to work with a subgroup H of smaller index and use higher degree factors in order to prove that $\mathrm{Gal}(f)$ is contained in a conjugate of U.

Example 3.5. *Let $f \in \mathbb{Z}[X]$ be a polynomial with Galois group 19T5 $\cong C_{19} \rtimes C_9$. This is a maximal subgroup of $\mathbf{A}_{19}$ of index $17!$. Since $19T5$ is not 2-transitive, take $S := \mathrm{Stab}_{\mathbf{A}_{19}}([1,2])$ the intersection of the point stabilisers of 1 and 2, take $F := x_1 - x_2$ and compute the resolvent R_F. This is a polynomial of degree $19 \cdot 18$ which has $\alpha_i - \alpha_j$ as roots for $1 \le i \ne j \le 19$ (assuming that those roots are different).*

Furthermore, using resultants, R_F can be computed symbolically without explicit knowledge of S or F. Factorisation of R_F finds two factors of degree 171. When we apply Theorem 3.2 we directly descend to the correct Galois group.

Let α be a root of the polynomial f in the last example. The factorisation approach of the last example is equivalent to the fact that $f/(x-\alpha) \in \mathbb{Q}(\alpha)[x]$ factorises into two degree 9 factors.

Very often the factorisation approach is not optimal or even feasible since the degree of the resolvent polynomial is too high for efficient factorisation. In some situations our algorithm produces a Galois group (as an actual permutation group on the roots) which is only correct with a very high probability. In this situation we can turn to the factorisation method in order to check our result, i.e. to give a proof that the result is mathematically correct. Since we assume knowledge of the action on the roots, it is not necessary to factor the resolvent polynomial. By analysing the proof of Theorem 3.2 we can write down the factor and check if it is in $\mathbb{Z}[X]$ and whether it divides the resolvent polynomial. Similar ideas were used by Casperson and McKay [2] to obtain polynomials with Galois group M_{11}.

Example 3.6. *Let p be a prime number, $G := \mathbf{S}_{p+1}$, and $H := \mathrm{PGL}(2,p) \le G$. Then $(G:H) = (p-2)!$ and H is a maximal subgroup. Furthermore G is sharply 3–transitive which means that for the resolvent method we have to use a polynomial acting on 4-sets of the roots which has degree $\binom{p+1}{4}$. For p=19 this polynomial has degree 4845 and splits into four factors of degree 570, 855, 1710, and 1710 resp. In our implementation we compute the Galois group using short cosets 7.1 with a very high probability. Then we use this group to approximate the degree 570 factor. When computing the corresponding stabiliser according to Theorem 3.2 we descend to G.*

We remark that the group $\mathrm{PSL}(2,p) \le \mathbf{A}_{p+1}$ is not 3–transitive. The 3-set polynomial, i.e. the resolvent corresponding to $S := \mathrm{Stab}_{\mathbf{A}_{p+1}}(\{1,2,3\})$ gives no information since this polynomial stays irreducible. But if we take the pointwise stabiliser $S := \mathrm{Stab}_{\mathbf{A}_{p+1}}[1,2,3]$ (intersection of the 3 pointwise stabilisers) then we will find 3 factors. The latter polynomial has only degree $(p+1)p(p-1)$ compared to $\binom{p+1}{4}$.

As a final example in this section we consider a degree 40 polynomial with Galois group PGsp(4,3). This polynomial was computed by Tim Dokchitser and for reasons of space we do not give the actual polynomial here. The Galois group is primitive and not 2–transitive. Furthermore this group is maximal in $\mathbf{A}_{40}$. The algorithm outlined

below, using only linear factors computes the Galois group within 130 seconds. However, factoring a suitable resolvent for the stabiliser of 2-sets, completes the computation in only 23 seconds.

4. Generic Invariants

We fix two groups $H < G \leq \mathbf{S}_n$ and assume unless explicitly stated otherwise, that H is a maximal subgroup of G. The aim of this section is to find a G-relative H-invariant polynomial $F \in \mathbb{Z}[\underline{X}]$ of small degree and a small number of terms. While the first aim can be obtained easily, the second is more difficult, and will be discussed later. To simplify notation we will write $\sum A$ to mean $\sum_{a\in A} a$ for suitable sets A, usually orbits.

The first observation is that it is always easy to write down some invariant. Certainly, every $\mathbf{S}_n$-relative invariant is G-relative as well for $H \leq G \leq \mathbf{S}_n$.

Lemma 4.1.

$$F := \sum_{\sigma\in H} (\prod_{i=1}^{n-1} X_i^i)^\sigma$$

is a $\mathbf{S}_n$-relative H-invariant.

While Lemma 4.1 proves the existence of G-relative H-invariant polynomials it is a very expensive invariant from the point of view of evaluation. Even assuming that the powers of the evaluation points are stored, the evaluation of each term needs $n-2$ multiplications, so that in total $\#H(n-2)$ multiplications are necessary. In order to improve on this we make use of the following well known facts ([5]):

Theorem 4.2. *For any polynomial $I \in \mathbb{Z}[\underline{X}]$, and every group H we have that $F(\underline{X}) := \sum\{I^h(\underline{X}) \mid h \in H\} =: \sum I^H(\underline{X})$ is H-invariant.*

For every H-invariant polynomial $F \in \mathbb{Q}[\underline{X}]$, there exist monomials m_i and coefficients $a_i \in \mathbb{Q}$ such that

$$F = \sum_{i=1}^{r} a_i \sum m_i^H.$$

Thus invariants of the form $\sum m^H$ form a vector space basis for the ring of all invariants.

The invariant ring $\mathbb{Q}[\underline{X}]^H$ of H invariant polynomials is a graded $\mathbb{Q}$-vector space. The dimensions of the summands can be read of the Hilbert-series:

$$f_H(t) := \sum_{i=0}^{\infty} t^i \dim(R_H)_i$$

where $(R_H)_i = \{r \in \mathbb{Q}[\underline{X}]^H \mid \deg r = i\} \cup 0$.

The Hilbert series can be computed from the knowledge of the classes of H *under conjugation:*

$$f_H(t) = \frac{1}{\#H} \sum_{c \in C} \frac{\#c}{\prod_{i=1}^{l} (1 - x^{c_i})^{d_i}},$$

where (c_i, d_i) is the cycle structure of any representative of the class c of H.

To improve on Lemma 4.1 we will try to find a small invariant as a basis element for some $(R_H)_d$ for d as small as possible. Unfortunately, there are pairs of groups, $G = \mathbf{S}_n$, $H = \mathbf{A}_n$ for example, where the invariant in Lemma 4.1 can be shown to be of minimal degree.

In the remainder of this section we will develop methods to compute a basis for $(R_H)_d$ the vector space of H-invariant polynomials of degree d and also for the subspace of G-relative polynomials. Our strategy will be to first compute a basis for the $\mathbf{S}_n$-invariant polynomials and then show how to refine this basis. We start with two observations:

Remark 4.3. (1) *Let* $F \in \mathbb{Z}[\underline{X}]$ *be a polynomial and* $H \leq \mathbf{S}_n$ *be a group. Then*

$$\sum_{\sigma \in H // \operatorname{Stab}_H(F)} F^\sigma = \sum F^H$$

and thus is H*-invariant.*

(2) *Let* $m = \prod_{i=1}^{n} X_i^{a_i}$ *be a monomial. Then we have*

$$\operatorname{Stab}_H(m) = \bigcap_{a \in \{a_i \mid 1 \leq i \leq n\}} \operatorname{Stab}_H(\{i \mid a_i = a\}),$$

thus stabilisers of monomials can be computed as intersections of stabilisers of points or sets. Of course, for $H = \mathbf{S}_n$ *those stabilisers can be made explicit as direct products of suitable* $\mathbf{S}_m$ *for* $m < n$*:*

(3) *Let* $\{1, \ldots, n\} = \cup_{i=1}^{r} A_i$ *be a partition. Then*

$$\operatorname{Stab}_{\mathbf{S}_n}(A_1, \ldots, A_r) \cong \prod_{i=1}^{r} \mathbf{S}_{A_i}.$$

4.1. **$\mathbf{S}_n$-invariant polynomials.** In this section we will develop methods to compute a basis of $\mathbf{S}_n$-invariant polynomials as well as indicate how to improve on the general method if we want to aim for relative invariants only. The algorithm presented here is similar to the ideas presented in [1, 11].

The key idea here is that the orbit sum

$$\sum_{\sigma\in\mathbf{S}_n//\operatorname{Stab}(f)} f^\sigma$$

does not depend on the representative f of the full orbit $f^{\mathbf{S}_n}$ and that the action of the group on some monomial m only depends on the partition of $\{1,\dots,n\}$ induced by $m=\prod_{i=1}^n X_i^{a_i}$:

$$\{1,\dots,n\}=\dot{\bigcup}_{a\in\{a_i\mid 1\le i\le n\}}\{i\mid a_i=a\}.$$

On the other hand, by giving a partition $\underline{A}:=\{A_i\mid i\}$ of $\{1,\dots,n\}$ and pairwise different integers $a_i\ge 0$ the orbit of $m(\underline{a},\underline{A}):=\prod_{i=1}^s\prod_{j\in A_i}X_j^{a_i}$ is uniquely defined by $\underline{A}$ already. Thus to solve our problem of finding a basis for $(R_{\mathbf{S}_n})_d$ we simply need to find all partitions and exponents such that $\sum_{i=1}^s a_i\#A_i=d$. We summarise this in an algorithm:

Algorithm 4.4. *Let d be an integer. We want to find a basis for $(R_{\mathbf{S}_n})_d$.*

(1) *Let $I:=\{\}$.*
(2) *Compute the set P of all partitions of d of length at most n.*
(3) *For $p\in P$ do*
(4) *Let $p=(p_1,\dots,p_i)$. Append I by $\prod_{j=1}^i X_j^{p_j}$*
(5) *(end of the for $p\in P$ loop)*

However, since we are eventually only interested in finding minimal degree invariants we introduce more reductions here. The operation of $\mathbf{S}_n$ on $m(\underline{a},\underline{A})$ does only depend on $\underline{A}$, so for a minimal degree invariant we can also stipulate that $a_i+1=a_{i+1}$ - otherwise the same behaviour can be obtained with smaller exponents. Similarly, minimal examples will be such that $\#A_1\ge\#A_2\ge\ldots\ge\#A_s$. As an example: the orbits of $X_1X_2^3$ and of $X_1X_2^2$ are essentially the same, namely the orbit of $\{\{1\},\{2\}\}$. Thus if we are looking for examples of minimal degree, then $X_1X_2^3$ need not be considered.

4.2. **H-invariants.** Let H be a subgroup of $\mathbf{S}_n$ and I be a set of monomials generating different $\mathbf{S}_n$-invariants via orbit sums. Here we address the problem of refining I to contain a (maximal) set of monomials generating H-invariants.

Let m be a monomial and $S:=\operatorname{Stab}_G(m)$ its stabiliser in some group $\mathbf{S}_n\ge G\ge H$. We use the following theorem

Theorem 4.5. *Let $G\ge H$ be groups, m a monomial and $S=\operatorname{Stab}_G(m)$ its stabiliser. Furthermore, let $S\setminus G/H$ be the double cosets of G with respect to S and H and let $\{g_i\mid 1\le i\le r\}$ be a set of representatives*

(i.e., $G = \cup_{i=1}^{r} Sg_iH$ and $Sg_iH = Sg_jH$ if and only if $i = j$). Then $\{m_i^g \mid 1 \le i \le r\}$ generate linearly independent H-invariants.

Proof. The linear independence is a direct consequence from the fact that the double coset decomposition induces a decomposition of m^G into pairwise disjoint H-orbits. □

Thus the computation of H-invariants is reduced to the computation of $\mathbf{S}_n$-invariants followed by a double coset decomposition. While in general double coset decompositions are hard to compute, it is feasible here. We make use of the ladder-technique of [18]: Usually to compute double cosets, one computes a coset decomposition with respect to one group and lets the other group act on them, thus the complexity depends on the size of the index of the larger group in G. This procedure is frequently helped by computing a descending chain from $G =: S_0 > \cdots > S_j = S$ down to one smaller group, S for example. The action of H on $S \setminus G$ can then be deducted from the action of H on $S_{i+1} \setminus S_i$. Unfortunately, it is hard and frequently impossible to find good subgroup chains, that is chains with small indices. The new idea introduced in [18] is to use a ladder rather than a chain, i.e. to allow up-ward steps as well as down-ward ones. In order to use this technique we therefore have to construct a suitable ladder. This will be achieved by the following procedure:

Algorithm 4.6. *Let G be a permutation group acting on Ω and $A \subseteq \Omega$ be arbitrary. This algorithm will compute a ladder G_i such that $G = G_0$, $G_r = \mathrm{Stab}_G(A)$ and if $G_i < G_{i+1}$ then $\#G_{i+1}/\#G_i \le \#\Omega$ and $G_i > G_{i+1}$ with $\#G_i/\#G_{i+1} \le \#\Omega$ otherwise.*

(1) *Let $B := \{\}$, and $i := 1$, $G_0 := G$.*
(2) *for $a \in A$ do*
(3) *add a to B and compute $G_i := \mathrm{Stab}_{G_{i-1}}\{a\}$.*
(4) *if $B \ne \{a\}$ then $G_{i+1} := \mathrm{Stab}_G B$ and set $i := i+2$. Otherwise set $i := i+1$.*

Proof. Let $A := \{a_1, \dots, a_n\}$, the properties of the G_i are direct consequences of the following facts:

(1) We either have $G_i = \mathrm{Stab}_{G_{i-1}}\{\{a_1, \dots, a_s\}, \{a_{s+1}\}\}$ (in which case $G_i < G_{i+1}$) or
(2) we have an up-ward step and obtain $G_{i+1} = \mathrm{Stab}_G\{a_1, \dots, a_{s+1}\}$. Note that $\mathrm{Stab}_{G_{i+1}}\{a_{s+1}\} = G_i$.
(3) For $G = \mathbf{S}_n$, we have $\#\,\mathrm{Stab}_G\{a_1, \dots, a_s\} = s!(n-s)!$ and $\#\,\mathrm{Stab}_G\{\{a_1, \dots, a_s\}, \{a_{s+1}\}\} = s!1!(n-s-1)!$.

(4) In general, $\mathrm{Stab}_G A = G \cap \mathrm{Stab}_{\mathbf{S}_n} A$, and for any groups $V < U < \mathbf{S}_n$ we have $(U \cap G : V \cap G) \leq (U : V)$, thus the bound on the indices follows.

□

For more general partitions, the Algorithm 4.6 will be called repeatedly:

Algorithm 4.7. *Let G be a permutation group acting on Ω and $A = \{A_1, \ldots, A_s\}$ a partition of Ω. This algorithm will compute a ladder G_i such that $G = G_0$, $G_r = \mathrm{Stab}_G(A)$ and if $G_i < G_{i+1}$ then $\#G_{i+1}/\#G_i \leq \#\Omega$ and $G_i > G_{i+1}$ with $\#G_i/\#G_{i+1} \leq \#\Omega$ otherwise.*

(1) *Let $U := G$.*
(2) *For $a \in A$ do*
(3) *Compute a ladder from U to $\mathrm{Stab}_U a$ using Algorithm 4.6 and print it.*
(4) *Let $U := \mathrm{Stab}_U a$.*

Let $G \leq \mathbf{S}_n$ be arbitrary and $H < G$ a maximal subgroup. In order to compute G-relative H-invariants, we now use one of the following algorithms:

Algorithm 4.8. *Let $H < G$ be as above and $d > 0$ be an integer. This algorithm will find a basis for the space of G-relative H-invariants of degree d.*

(1) *Compute a basis B for $(R_{\mathbf{S}_n})_d$ using 4.4.*
(2) *For each $b \in B$ do*
(3) *Compute the corresponding partition A*
(4) *Use 4.7 to compute a ladder L from $\mathbf{S}_n$ to $\mathrm{Stab}_{\mathbf{S}_n}(b)$ using the partition A.*
(5) *Use L to compute a set C of double coset representatives for $\mathrm{Stab}_{\mathbf{S}_n} b \setminus \mathbf{S}_n / H$*
(6) *For each $c \in C$ do*
(7) *Compute the indices of the stabilisers $(H : \mathrm{Stab}_H b^c)$ and $(G : \mathrm{Stab}_G b^c)$. If they differ then b^c generates a G-relative H-invariant. In this case print $\sum_{h \in H // \mathrm{Stab}_H b^c} b^{ch}$*

The correctness of the algorithm follows immediately from the above discussions. We remark that if we want only one invariant rather than a basis, we can use a probabilistic approach:

Algorithm 4.9. *Let $H < G$ as above and $d > 0$ be an integer such there exists an G-relative H-invariant of degree d. This algorithm will find one G-relative H-invariant of degree d.*

(1) *Compute a basis* B *for* $(R_{\mathbf{S}_n})_d$ *using 4.4.*
(2) *repeat*
(3) *Compute a random element* $\sigma \in \mathbf{S}_n$
(4) *For each* $b \in B$ *check if* $(G : \mathrm{Stab}_G\, b^\sigma)$ *differs from* $(H : \mathrm{Stab}_H\, b^\sigma)$. *If so, print* $\sum_{h \in H// \mathrm{Stab}_H b^\sigma} b^{\sigma h}$ *and terminate.*

To find a (minimal) degree d such that there exists an G-relative H-invariant we simply compute the difference of the Molien series $f_H(t) - f_G(t) = \sum_{i=1}^\infty s_i t_i$ and take d as the index of any non zero coefficient.

5. Special Invariants

Unfortunately, there are examples where the generic invariants are too expensive to compute or the given presentation needs too many arithmetic operations to evaluate the invariant. The best known example for this are the groups $H = A_n$ and $G = \mathbf{S}_n$. Clearly, H is a maximal subgroup of G and the invariant

$$F_1(X_1, \ldots, X_n) := \sum_{\sigma \in H} (\prod_{i=1}^{n-1} X_i^i)^\sigma$$

given in Lemma 4.1 is a $\mathbf{S}_n$-relative A_n-invariant polynomial of smallest possible total degree. If we store the powers of X_i we need $(n-2)n!/2$ multiplications in order to evaluate this invariant. If the characteristic is not equal to 2, then a better invariant is well known: (for any $\sigma \notin H = \mathbf{A}_n$)

$$F_2(X_1, \ldots, X_n) := \prod_{1 \le i < j \le n} (X_i - X_j) = F_1 - F_1^\sigma,$$

which can be evaluated using $n(n-1)/2$ multiplications, if the factored form is used.

Most of the special invariants presented here follow the same pattern and are derived from the same source, namely from the different action of G and H on natural objects like the action on blocks or block systems. Ultimately, as we saw above in the discussion of general factorisation patterns, we can use permutation presentations for G acting on the cosets by any subgroup $V < G$.

In the following we assume that $H < G \le \mathbf{S}_n$ are acting on $\Omega := \{X_1, \ldots, X_n\}$. Let us start with the case that H is acting intransitively. The proof of the following lemma is trivial.

Lemma 5.1. *Assume that there exists an orbit $\emptyset$ of H which is not invariant under G. Then*

$$F(X_1, \ldots, X_n) := \sum_{X_i \in \emptyset} X_i \tag{1}$$

is an H-invariant G-relative polynomial.

We remark that intransitive groups may occur in our applications even if we start with transitive groups. The reason is that some of the following algorithms will reduce the problem recursively to groups of smaller degree.

Let us assume for the rest of the section that the given groups $H \leq G \leq \mathbf{S}_n$ are transitive. For transitive groups the notion of blocks and block systems are very important. We remark that most of the following invariants are well known, e.g. see [9, 10].

Definition 5.2. *Let $G \leq \mathbf{S}_n$ be transitive and $\emptyset \neq B \subseteq \Omega$ be a subset. Then B is called a block, if for all $g \in G$ we have $B^g \cap B := \{X^g \mid X \in B\} \cap B \in \{\emptyset, B\}$. Blocks of size 1 and n are called trivial blocks.*

It is very easy to see that B^g is a block if B is a block. By acting on a block B we get a partition of Ω which is called block system. Therefore every block is contained in a block system. Furthermore it is easy to see that the blocks containing X_1 are in 1-1 correspondence to the groups $G_{X_1} \leq U \leq G$, where $G_{X_1} = \mathrm{Stab}_G\{1\}$ is the point stabiliser of G and U is the stabiliser of the block, i.e. $U = \mathrm{Stab}_G B = \{g \in G \mid B^g = B\}$.

If $H \leq G$ then clearly every block(-system) of G is a block(-system) of H. But it may be the case that H possesses more blocks.

Lemma 5.3. *Let $H \leq G \leq \mathbf{S}_n$ be transitive groups and assume that $B_1, \ldots, B_m$ is a block system of H, but not one of G. Then*

$$F(X_1, \ldots, X_n) := \prod_{i=1}^{m} \sum_{X \in B_i} X \tag{2}$$

is an H-invariant G-relative polynomial.

Proof. Every $h \in H$ only permutes the factors of F and therefore stabilises F. Let $g \in G \setminus H$. Then there exist X_i and X_j lying in the same block which are mapped to different blocks. This produces a monomial of F^g containing $X_i X_j$ which does not exist in F. Since cancellations are impossible, we get the desired result. □

Now we can assume that the block systems of H and G coincide. Now let $B_1, \ldots, B_m$ be a block system of H (and G). We can define two canonical actions of G and H. One is by simply permuting the

blocks which give transitive permutation representations $\bar{G}$ and $\bar{H}$ on m points. We get the following exact sequences of groups:

$$1 \to N_G \to G \to \bar{G} \to 1,\ 1 \to N_H \to H \to \bar{H} \to 1$$

where N_G (resp. N_H) is the kernel of the permutation representation.

In the case that $N_H = N_G$ we can apply the following lemma. We remark that we always get $N_H = N_G$ if $\bar{H} \neq \bar{G}$ and we assume that H is a maximal subgroup of G.

Lemma 5.4. *Let $H \leq G \leq \mathbf{S}_n$ be transitive groups with a common block system $B_1, \ldots, B_m$. Assume that the above defined normal subgroups N_H and N_G are equal. Let $E(X_1, \ldots, X_m)$ be an $\bar{H}$-invariant $\bar{G}$-relative polynomial. Then*

$$F(X_1, \ldots, X_n) := E(Y_1, \ldots, Y_m) \text{ for } Y_i := \sum_{X \in B_i} X \tag{3}$$

is an H-invariant G-relative polynomial.

Proof. Elements of $N_H = N_G$ only change the ordering of the sum defining Y_i. Therefore an element g acts on F via the action of $\bar{g}$ on E. Therefore the polynomial F is H-invariant. In order to show the G-relativity, we need to prove that for $g \in G \setminus H$ we have $\bar{g} \notin \bar{H}$. The last statement easily follows from $N_H = N_G$. □

The other action can be defined within a block B_1 via $\mathrm{Stab}_G(B_1)|_{B_1}$. We get the following invariant.

Lemma 5.5. *Let $H \leq G \leq \mathbf{S}_n$ be transitive groups with a common block system $B_1, \ldots, B_m$. Let $\tilde{H} := \mathrm{Stab}_H(B_1)|_{B_1}$, $\tilde{G} := \mathrm{Stab}_G(B_1)|_{B_1}$ and assume $[G : H] = [\tilde{G} : \tilde{H}]$. Let $E(X_{i_1}, \ldots, X_{i_l})$ where $B_1 = \{X_{i_1}, \ldots, X_{i_l}\}$ be an $\tilde{H}$-invariant $\tilde{G}$-relative polynomial. Furthermore let $\{\sigma_1, \ldots, \sigma_m\}$ be a system of representatives of left cosets of $\mathrm{Stab}_H(B_1)$ in H.*

Then $F := E^{\sigma_1} + \cdots + E^{\sigma_m}$ is an H-invariant G-relative polynomial.

Proof. An element of H can be uniquely written as a product of an element of $\mathrm{Stab}_H(B_1)$ and some σ_i. The first one stabilises E and the second one only permutes the E^{σ_i}. Therefore F is invariant under H. Since $[G : H] = [\tilde{G} : \tilde{H}]$ we see that $\{\sigma_1, \ldots, \sigma_m\}$ are representatives of the left cosets of $\mathrm{Stab}_G(B_1)$ in G. Since an element $g \in G \setminus H$ can be uniquely written as a product $\tilde{g}\sigma_i$ of an element $\tilde{g} \in \mathrm{Stab}_G(B_1)$ and some σ_i we get that the element $\tilde{g}$ cannot be an element of $\mathrm{Stab}_H(B_1)$. Therefore $E^{\tilde{g}} \neq E$. Furthermore the X_j which appear in E^{σ_i} are different for different i's which shows that $F^g \neq F$. □

Now we have to deal with groups where the number of block systems is the same and it is not possible to use Lemma 5.4 or Lemma 5.5. In this situation, we can try the following ([9, 6.19]):

Lemma 5.6. *Let* $U := \mathrm{Stab}_H(B_1)|_{B_1} = \mathrm{Stab}_G(B_1)|_{B_1}$ *and* $K_1 < K_2 \leq U$*. Now let* F *be a* K_2*-relative* K_1*-invariant such that* $O := F^G = F^H$ *is not too large, finally let* $\rho : G \to \mathbf{S}_O$ *be the permutation representation of* G *on* O*. If* $\rho(H) \neq \rho(G)$ *then let* Y *be a* $\rho(G)$*-relative* $\rho(H)$*-invariant. For a suitable Tschirnhausen transformation* $t \in \mathbb{Z}[x]$ *we have that* ($F^G = \{F^{\sigma_1}, \ldots, F^{\sigma_o}\}$)*:*

$$I := Y(t(F^{\sigma_1}(\underline{X})), \ldots, t(F^{\sigma_o}(\underline{X})))$$

is a G*-relative* H*-invariant.*

Proof. Since G and H act identically on the block B_1, the orbits F^G and F^H are the same. By construction, I is clearly H-invariant, all that we need to show is that I is not G-invariant. Since F is not $\rho(G)$ invariant, this is immediate. □

It should be noted that the use of blocks above is only part of the attempt to create an invariant F with a small orbit.

The following theorem is a generalisation of a result of Eichenlaub [7], who proved the corresponding result for wreath products of symmetric groups. We remark that a wreath product $U \wr V$ is a semidirect product of the type $U^m \rtimes V$, where $V \leq S_m$ and the action of V permutes the copies of U. For a formal definition we refer the reader to [6, p. 46].

Theorem 5.7. *Let* $G = U \wr V$ *be the wreath product acting on* $X_{i,j}$ ($1 \leq i \leq d, 1 \leq j \leq m$)*, where* $U \leq S_d$*,* $V \leq S_m$ *and* $md = n$*. Furthermore let* $N \trianglelefteq U$ *be a normal subgroup of index 2. Let* E *be an* N*-invariant* U*-relative polynomial with the property that* $E^u = -E$ *for all* $u \in U \setminus N$*. Denote by* s_k *the* k*-th elementary symmetric function on* m *letters. Then* G *has a subgroup* H *of index 2 and*

$$F(X_{1,1}, \ldots, X_{d,m}) := s_m(d_1, \ldots, d_m) = d_1 \cdots d_m$$

is an H*-invariant* G*-relative polynomial, where* $d_j := E(X_{1,j}, \ldots, X_{d,j})$*.*

We remark that in the original statement given in [7] there are two other subgroups of index 2. One is $S_d \wr A_m \leq S_d \wr S_m$ which can be dealt with Lemma 5.4 and the other one comes from the fact that whenever we have two subgroups of index 2, there will be a third one. An invariant for this can be efficiently computed using the first two invariants, see Lemma 5.8.

Proof. Clearly, we have $N \wr V \leq U \wr V$ and using Lemma 5.5 we get that $E + E^u$ with $u \in U \setminus N$ is an $N \wr V$-invariant G-relative polynomial. Let

$u \in U \setminus N$ be an arbitrary element and let u_1 and u_2 be the canonical images of u in the first and second copy of U^m in G, respectively. Now we claim that $H = \langle N \wr V, u_1u_2 \rangle \leq G$. Clearly, F fixes all elements of $N \wr V$ because all d_i are fixed by elements of N and swapped by elements of V. The element u_1u_2 fixes $d_3, \ldots, d_m$ and has the property that $d_1^{u_1u_2} = -d_1$ and $d_2^{u_1u_2} = -d_2$. Therefore we get $F^{u_1u_2} = F$. For an arbitrary element $g \in G$ we get that $F^g = \pm F$ and therefore the index of H in G is at most 2. Clearly, $F^{u_1} = -F$ and therefore $H \neq G$ and F is G-relative. □

We remark that this invariant can be applied to groups G which are not a wreath product. E.g. it could be possible that G is contained in a wreath product, but not in the index 2-subgroup and H is contained in that index 2-subgroup.

As already mentioned it is possible to combine relative invariants in order to get new ones. The following lemmata all work in this generic situation. Suppose $G \leq \mathbf{S}_n$ has two subgroups $H_1 < G$ and $H_2 < G$ with G-relative invariants F_i. On the invariant field side, this corresponds to $\mathbb{Q}(\underline{X})^G$ having two finite separable extensions $\mathbb{Q}(\underline{X})^G(F_i)$ corresponding to $\mathbb{Q}(\underline{X})^{H_i}$ with normal closures M^{C_i} and $C_i := \mathrm{Core}_G(H_i)$. In this situation we can transfer information about H_i to all subfields (and the corresponding fix groups) of the compositum $M^{C_1}M^{C_2} = M^{\mathrm{Core}_G(H_1 \cap H_2)}$.

The first such example already appears in [7].

Lemma 5.8. *Let $G \leq \mathbf{S}_n$ be a permutation group which has two subgroups $H_1 \neq H_2$ of index 2 with G-relative H_i-invariants F_i. If $F_i^g = \pm F_i$ for $g \in G$, then there exists a third subgroup $H_3 := (H_1 \cap H_2) \cup ((G \setminus H_1) \cap (G \setminus H_2))$ of G such that F_1F_2 is a G-relative H_3-invariant.*

Proof. An element of $H_1 \cap H_2$ clearly stabilises F. Therefore let $h \in H_3 \setminus H_1 \cap H_2$. Then $h \notin H_1 \cup H_2$ and therefore $F_1^h = -F_1$ and $F_2^h = -F_2$ which gives $F^h = F$. This proves that F is H_3-invariant. Let $g \in G \setminus H_3$. Then $g \in H_1$ or $g \in H_2$, but $g \notin H_1 \cap H_2$. Therefore $F^g = -F$ and F is G-relative. □

Even if the invariants do not satisfy $F_i^g = \pm F_i$, the above Lemma 5.8 can be used, since for $G//H_i = \{\mathrm{Id}, g\}$ we see that $\tilde{F}_i := F_i - F_i^g$ is a G-relative H_i-invariant with the desired property $\tilde{F}_i^g = -\tilde{F}_i$.

In the more general situation we can still use the field theoretic view to combine information from two (or more) subgroups: Assume $G < \mathbf{S}_n$ has two subgroups $H_i < G$ $(i = 1, 2)$ with G-relative H_i-invariants F_i, set $H_{12} := H_1 \cap H_2$ and $C_i := \mathrm{Core}_G(H_i)$ for $i \in \{1, 2, 12\}$. Then for

any maximal subgroup $C_{12} < H_3 < G$ an G-relative H_3 invariant can be constructed by any of the following methods from F_i, $(i = 1, 2)$ and an G/C_{12}-relative H_3/C_{12} invariant. Also, set $K := \mathbb{Q}(\underline{X})^G$.

(1) (Intransitive construction) Let $\tilde{H}_i$ be the permutation representation of G on $G//H_i$, $i \in \{1, 2, 12\}$. We consider the subdirect product $\tilde{H}_1 \times_{H_{12}} \tilde{H}_2 \cong \tilde{H}_{12} \cong G/C_{12}$. The maximal subgroup $H_3 < G$ corresponds to a maximal subgroup $\tilde{H}_3 < \tilde{H}_{12}$. Let F be an invariant for this pair, then $F([F_1^s : s \in G//H_1], [F_2^s : s \in G//H_2])$ is a G-relative H_3-invariant.
(2) (Transitive construction) By the primitive element theorem, we can find an invariant F_i, $i = 1, 2$ of degree 1 such that $K(F_i) = \mathbb{Q}(\underline{X})^{C_i}$. Again, by the primitive element theorem, we find some r such that $F_1 + rF_2$ is primitive for $\mathbb{Q}(\underline{X})^{C_{12}}$. From here it is straight forward to obtain an invariant for H_3 as a polynomial in $F_1 + rF_2$.

In general, this is only applicable if the indices of the participating groups are small. In particular the transitive construction is mainly of interest for normal subgroups.

6. Intransitive groups

The Stauduhar algorithm works for intransitive groups in the same way. Let $f \in \mathbb{Q}[x]$ be a squarefree polynomial of degree n. Assume that $f = f_1 \cdots f_r \in \mathbb{Q}[x]$ has r factors of degree n_i, respectively. Then we know that $\mathrm{Gal}(f)$ is a subgroup of the intransitive group $\mathbf{S}_{n_1} \times \ldots \times \mathbf{S}_{n_r} \leq \mathbf{S}_n$. Using the methods for irreducible polynomials we can compute the Galois groups $G_i := \mathrm{Gal}(f_i) \leq \mathbf{S}_{n_i}$. Then $\mathrm{Gal}(f) \leq G_1 \times \ldots \times G_r \leq \mathbf{S}_n$. This direct product can be used as a starting group of our algorithm.

In order to simplify the presentation, let us assume that $\mathrm{Gal}(f) \leq G_1 \times G_2 < \mathbf{S}_n$. This is no restriction, since we do not assume that G_1 or G_2 are transitive. Therefore we have a corresponding factorisation $f = f_1 f_2 \in \mathbb{Q}[x]$, where we do not assume that f_1, f_2 are irreducible. All groups H with $\mathrm{Gal}(f) \leq H \leq G_1 \times G_2$ have a special structure. Let us start to theoretically describe $\mathrm{Gal}(f)$. Denote by N_i the splitting field of f_i. Furthermore define N to be the compositum $N_1 N_2$ and $M := N_1 \cap N_2$. Let U be the Galois group of M/K. Then the Galois group of N/M is the subdirect product (fibre product) $G_1 \times_U G_2$ with common factor group U. Denote by $\phi_i : G_i \to U$ the corresponding epimorphisms. Then $G_1 \times_U G_2$ can be realized via

$$G_1 \times_U G_2 = \{(g_1, g_2) \in G_1 \times G_2 \mid \phi_1(g_1) = \phi_2(g_2)\}.$$

Furthermore we see that this is automatically a normal subgroup of $G_1 \times G_2$.

Now let us consider the case that $H = G_1 \times_U G_2$ and $G = G_1 \times G_2$. Define $V_i \leq G_i$ to be the normal subgroups such that $G_i/V_i = U$. Then we get the following chain of subgroups:

$$V_1 \times V_2 \leq G_1 \times_U G_2 \leq G_1 \times G_2.$$

A $V_1 \times V_2$-invariant $G_1 \times G_2$-relative invariant can be computed by using the corresponding V_i-invariant G_i-relative invariants defined on the components and the primitive field argument. This invariant can be improved to an G-relative H-invariant by taking sums over elements from $H//(V_1 \times V_2)$.

In general, since the generic invariants are computationally bad, we would like to use special invariants in this case as well. However, none of them work for intransitive groups, so our only chance here is to compute a transitive representation of the larger group and then test for special invariants in the transitive representation. Let $H < G$ and G be intransitive. If the G-orbits and the H-orbits differ, we get a trivial G-relative H-invariant from any H-orbit that is no G-orbit. Hence, we assume that the orbits are the same. Similarly, we assume that the action of G and H on the orbits agree. In this case we construct a transitive representation $\phi : G \to \mathbf{S}_T$ of G on the set $T := \prod_{o \in O} o$ where the product runs over all orbits. The image $\phi(H)$ of H under this representation is again a subgroup of $\phi(G)$ and we can now test for special invariants. Assume $I \in \mathbb{Z}[X_t \mid t \in T]$ is such an invariant. Then $I(y(\sum_{o \in O} X_{t_o}) \mid t \in T)$ for some suitable Tschirnhaus transformation $y \in \mathbb{Z}[x]$ is a G-relative H-invariant.

Proof. For any fixed $t \in T$ it is clear that $\sum_{o \in O} X_{t_o}$ is a primitive element for $(\mathbb{Q}[X_t \mid t \in T]^{\phi(H)}/\mathbb{Q}[X_t \mid t \in T]^{\phi(G)}$ since all the conjugates $\sum_{o \in O} X_{s_o}$ $s \in T$ are different. $\square$

7. Computation of Galois Groups

In the previous sections we investigated a variety of special and generic constructions for invariants. Here we are going to discuss how they can be used to compute Galois group.

To start, let f be an irreducible monic polynomial in $\mathbb{Z}[x]$ and let K be an extension of $\mathbb{Q}$ such that $f(x) = \prod_{i=1}^{n}(x - \alpha_i)$, i.e. K a fixed splitting field, not necessarily a minimal one. We want to compute

$$\mathrm{Gal}(f) := \mathrm{Aut}(\mathbb{Q}(\alpha_1, \ldots, \alpha_n)/\mathbb{Q}) \leq \mathbf{S}_{\alpha_1,\ldots,\alpha_n}.$$

Since we assume f to be irreducible, $\mathrm{Gal}(f)$ is going to be transitive.

Suitable choices for K are p-adic fields or the field of complex numbers. We will defer the choice of the field until we discussed the operations we need to perform with it. Thus we assume that (somehow) we are given a field K and the roots α_i in some arbitrary but fixed ordering.

The main algorithm will, starting with some group $G \geq \mathrm{Gal}(f)$ refine the initial guess by considering (maximal) subgroups. Hence the first step is a good starting group.

7.1. **Starting Group.** Naively, obviously, $\mathrm{Gal}(f) \leq \mathbf{S}_n$, so $G := \mathbf{S}_n$ is a valid start. However, this is very bad for the subsequent steps as $\mathbf{S}_n$ has maximal subgroups of very large index.

Set $E := \mathbb{Q}(\alpha_1) = \mathbb{Q}(x)/f$ which is a number field of degree n. Using algorithms developed by Klüners [14] (or recently Klüners, van Hoeij and Novocin [12]) it is relatively easy to find subfields $\mathbb{Q} \subset F \subset E$ - or to decide that there are no subfields. By Galois theory, the subfields are in 1-1 correspondence to the (unknown) block systems of the (unknown) Galois group. Thus the non-existence of subfields proves the group to be primitive.

Assuming we have a non-trivial subfield $F = \mathbb{Q}(\beta)$ for $\beta = \sum r_l \alpha_1^l$. Then $B_1 := \{\alpha_j \mid \sum r_l \alpha_i^l = \sum r_l \alpha_1^l\}$ is the block containing α_1, the other blocks are computed similarly. From here, it is trivial to compute the wreath product W_F corresponding to this block system, and $\mathrm{Gal}(f) \subseteq W_F$. The construction can be improved if we compute the Galois group of $F/\mathbb{Q}$ (and even more if we compute the group of F/E - but this is too expensive in practise). Doing this for all subfields, we compute a suitable starting group.

If there are no subfields, hence $\mathrm{Gal}(f)$ is primitive, we can try to obtain a good starting group from factoring suitable resolvent polynomials as indicated above.

7.2. **Stauduhar.** We assume now that we have some $\mathrm{Gal}(f) \leq G \leq \mathbf{S}_n$. The next step is now to either prove that G is the Galois group or replace it by some smaller group H. Now let $H < G$ be maximal. If we have subfields, we should also verify that H admits the same block systems as G, otherwise it cannot contain the Galois group.

We now find a G-relative H-invariant F using any of the methods above, verify that

$$F^\sigma(\underline{\alpha}) = F^\tau(\underline{\alpha}) \text{ if and only if } \sigma\tau^{-1} \in H \tag{4}$$

and compute

$$C := \{\sigma \in G//H \mid F^\sigma(\underline{\alpha}) \in \mathbb{Z}\}.$$

If C is non-empty, then $G := \cap_{\sigma \in C} H^\sigma$ will be our new group.

Remark 7.1. *Obviously, if $(G : H)$ is large, this is going to be very inefficient, if not impossible. If we have knowledge of some non-trivial element $\tau \in \mathrm{Gal}(f)$, coming from some known automorphism of K, then we can aid the computation of C. Instead of $G//H$ we only compute*

$$G//_\tau H := \{\sigma \in G//H \mid \tau \in H^\sigma\}$$

the so called short-coset. The actual computation of $G//_\tau H$ can be performed even if $(G : H)$ is too large to be computed [10, 4.6]

However, we do not know how to test (4) effectively. All we do here is to apply some probabilistic test, i.e. test for some 100 cosets if the images differ and rely on an independent proof later to justify the result.

7.3. **Splitting Field.** Now that we have looked at the components of the algorithm, we can discuss the splitting field. As we saw, we need to be able to quickly evaluate $F(\underline{\alpha})$, decide if two such evaluations are different and, finally, test if $F(\underline{\alpha}) \in \mathbb{Z}$. All of those tasks would be trivial if we could use an purely algebraic, exact representation of a splitting field K, however, since $[K : \mathbb{Q}] \geq \#\,\mathrm{Gal}(f)$ this is in general not practical. Using $K = \mathbb{C}$ as Stauduhar did is possible, but makes it difficult to decide if $F(\underline{\alpha}) \in \mathbb{Z}$, this would involve a careful analysis of the numerical properties of F. By restricting the invariants to be free of division, and using a suitable p-adic field K, we can overcome most problems, although we actually need both complex and p-adic information. We choose a suitable prime p and compute a finite extension K of $\mathbb{Q}_p$. The complex information is used to derive the p-adic precision necessary to guarantee correctness. Let $0 < M$ be such that $|\alpha_i| \leq M$ for all complex roots α_i. It is now easy to compute N such that $|F^\sigma(\underline{\alpha})| \leq N$ - for all σ as we cannot align the ordering of complex and p-adic roots. Thus using a p-adic precision k such that $p^k > 2N$ means that we can easily find (the unique) $R \in \mathbb{Z}$ such that $F(\underline{\alpha}) = \theta \bmod p^k$ and $|R| < N$.

To proof $F(\underline{\alpha}) = \theta$ is equivalent to showing that $R_F(\theta) = 0$. Since $R_F \in \mathbb{Z}[t]$ and $\theta \in \mathbb{Z}$, we have $R_F(\theta) \in \mathbb{Z}$. From $F(\underline{\alpha}) = \theta \bmod p^k$ we get $p^k | R_F(\theta)$, while on the other hand $|R_F(\theta)| \leq (|\theta| + N)^{(G:H)}$, so either $R_F(\theta) = 0$ or

$$p^k \leq (|\theta| + N)^{(G:H)}.$$

so we can easily compute k large enough to proof $R_F(\theta) = 0$. Unfortunately, $k = O(G : H)$, so k is too large to be useful in general. Similarly to the use of short cosets 7.1 we apply a hybrid approach. We choose

k large enough to *find* θ, i.e. $k = O(\log N)$ and rely on a final proof step to verify the computation.

7.4. **Proof.** We assume we have a subgroup chain $G =: H_0 > H_1 > \ldots > H_r$ where we know that $\mathrm{Gal}(f) \subset H_0$ and some of the steps are proven (if the index $(H_i : H_{i+1})$ is small enough, the above method will prove that $\mathrm{Gal}(f) \subset H_{i+1}$ if $\mathrm{Gal}(f) \subset H_i$) while others are not. Following [9], we find indices $i \neq j$ such that provably $\mathrm{Gal}(f) \subset H_i$ and probably $\mathrm{Gal}(f) \subset H_j$. We then look for resolvent polynomials R coming from point-set stabilisers to verify this step: Assuming that $\mathrm{Gal}(f) \subset H_j$, there has to be a factor $I|R$ that does not exist if $\mathrm{Gal}(f) = H_i$.

7.5. **Reducible Polynomials.** Most of the outlined method applies to reducible polynomials as well, the key difference is that the groups occurring are naturally intransitive, which excludes most of the special invariants. Let $f = \prod f_i$ be squarefree and monic. We start by fixing a common splitting field K, compute the roots of f and compute the Galois groups $G_i = \mathrm{Gal}(f_i) \subset \mathbf{S}_{\underline{\alpha}}$. Galois theory now states that $\mathrm{Gal}\, f < \prod G_i$, so we re-start the algorithm above with f and $G := \prod G_i$ as a starting group. We note that only subgroups $H < G$ need to be investigated that project onto the full Galois groups of the factors, i.e. the final group is a subdirect product of the G_i.

8. Numerical Results

In order to test the procedure outlined in this paper, we applied it to the complete contents of a database of polynomials [16] with known Galois groups (http://galoisdb.math.upb.de). This database contains explicit examples, sometimes many, for most groups of degree ≤ 23. For more than 10^6 polynomials, a total of 4835 different Galois groups have been computed. In this range, for 4624 groups the average runtime was less than 5 seconds. Only 5 groups took more than 30 seconds to compute.

Looking at some example in more detail. Let

$$f := x^{20} - 308x^{16} + 33396x^{12} - 1554608x^8 + 28579232x^4 - 113379904$$

with Galois group $20T684$ of order 61440. We start by factoring f modulo several small primes to select $p = 89$ for our splitting field which is an unramified cubic extension of $\mathbb{Q}_{89}$. Next, subfields are computed, and we recurse by computing the Galois group of the degree 10 subfield first. Using the subfield data, we conclude that the Galois group of f is a subgroup of $20T992$ of order $2^{17} \cdot 3 \cdot 5$. This group has 6 maximal transitive subgroups, of which only one is a candidate for the target

groups, the others can be excluded by block systems or intersections with other known groups. The only group to test further is isomorphic to $20T807$ of index 2^4, for this pair of groups we construct a special invariant using 5.6. The group $20T807$ now has 8 maximal transitive groups, 2 of which we need to test further. For both subgroups, both isomorphic to $20T684$ of index 2, our algorithm fails to find special invariants, thus uses the generic ones from section 4. Unfortunately, on evaluation of those invariants, we detect duplicate values, hence have to resort to Tschirnhaus transformation. In this example, we end up trying up to 10 different transformations of degree up to 7 before we find one to remove the duplicate values, hence makes the resultant squarefree and a descent is found. The resulting group again has 4 maximal transitive subgroups, none of which however are possible, thus the computation terminates. The "long" runtime here is a result of the generic invariants one the one hand and the need for Tschirnhaus transformations on the other. By construction, the generic invariants chosen are of minimal degree but need $> 500,000$ multiplications for a single evaluation. Due partly to the Tschirnhaus transformations, a p-adic precision of > 60 digits is used which then explains the runtime.

Comparing this to other polynomials with the same group, we see that the runtime varies substantially (20 - 240 seconds) which is due to the number of Tschirnhaus transformations used: this depends on the polynomials and not (directly) on the group. In this example, the "nice" structure of the polynomial with lots of zero-coefficients indirectly causes the transformations, while we could "easily" fix this by a transformation of the original polynomial, this would also incur a drastic growth of the coefficients, thus rendering this mostly useless.

Overall, the runtime can be seen to depend mainly on the groups as this determines the invariants and the descent tree transversed. Long runtimes typically are the result of bad invariants (generic invariants, frequently if the groups are very similar, i.e. small index). Large index subgroups, while posing a potential problem for the verification, are frequently easy to compute with: the short cosets reduce the number of candidates dramatically and the vastly differing groups make finding of invariants easy.

9. Future Work

The algorithm, as presented here, has two major weaknesses: it needs to find "good" invariants and it "needs" a small index in order to have verifiable results. Thus more work is needed to increase the number of "special" invariants. In fact, work in this direction has already

commenced e.g. Elsenhans [8] found better invariants for pairs of intransitive groups and for certain (large) pairs of 2-groups. In order to address the verification problem, maybe the use of non-linear factors of the resolvent polynomials as demonstrated in 3.5 should be investigated further.

However, as of now, we have a degree independent complete algorithm to compute Galois groups of univariate polynomials. The algorithm is very efficient and has been used on polynomials of degree > 100 already.

References

[1] I. Abdeljaouad. Calculs d'invariants primitifs de groupes finis, *Theor. Inform. Appl.* **33** (1999), 59–77.

[2] D. Casperson. and J. McKay. Symmetric functions, m-sets, and Galois groups, *Math. Comput.* **63** (1994).

[3] H. Cohen, *A course in computational algebraic number theory.* Springer, Berlin, 1993.

[4] On transitive permutation groups, John H. Conway, Alexander Hulpke and John McKay, *LMS J. Computation and Mathematics*, **1**(1998), 1–8.

[5] H. Derksen and G. Kemper. Computational invariant theory, *Encyclopaedia of Mathematical Sciences. Invariant Theory and Algebraic Transformation Groups*, Springer, Berlin, 2002.

[6] J. Dixon and B. Mortimer. *Permutation groups.* Springer, Berlin-Heidelberg-New York, 1996.

[7] Y. Eichenlaub. *Problèmes effectifs de théorie de Galois en degrés 8 à 11.* Thèse, Université Bordeaux 1, 1996.

[8] A.-S. Elsenhans. *Invariants for the Computation of intransitive and transitive Galois Groups. J. Symb. Comput.* **47** (2012), 315–326.

[9] K. Geißler. *Berechnung von Galoisgruppen über Zahl- und Funktionenkörpern.* PhD thesis, Technische Universität Berlin, 2003.

[10] K. Geißler and J. Klüners. Galois group computation for rational polynomials, *J. Symb. Comput.* **30** (2000), 653–674.

[11] K. Girstmair. On invariant polynomials and their application in field theory, *Math. Comput.* **48** (1987), 781-797.

[12] M. van Hoeij, J. Klüners and A. Novocin. Generating Subfields, *ISSAC'2011*, 345–352.

[13] G. Kemper and A. Steel. Some Algorithms in Invariant Theory of Finite Groups, in *Computational Methods for Representations of Groups and Algebras, Euroconference in Essen, April 1-5 1997*, edited by P. Dräxler, G.O. Michler and C.M. Ringel, Birkhäuser, Progress in Mathematics,173, 1997.

[14] J. Klüners. Über die Berechnung von Automorphismen und Teilkörpern algebraischer Zahlkörper, *PhD TU-Berlin* (1997).

[15] J. Klüners and G. Malle. Explicit Galois realization of transitive groups of degree up to 15, *J. Symb. Comput.* **30** (2000), 675–716.

[16] J. Klüners and G. Malle. A database for field extensions of the rationals, *LMS J. Comput. Math.*,**4** (2001), 182–196.

[17] Rüdiger Loos. Computing in algebraic extensions, Symbolic and Algebraic Computation, Springer, Wien, 1982, pp. 173–187.
[18] B. Schmalz. Verwendung von Untergruppenleitern zur Bestimmung von Doppelnebenklassen, *Bayreuther Mathematische Schriften* **31** (1990), 109–143.
[19] L. Soicher. The Computation of Galois Groups, *M.Comp.Sci. Thesis*, Concordia University, Montreal, 1981. `http://www.maths.qmul.ac.uk/~leonard/mcompsc_soicher.pdf`
[20] R. Stauduhar. The Determination of Galois Groups, *Math.Comput.* **27** (1973), 981–996.
[21] K. Yokoyama and G. Renault. A Modular Method for Computing the Splitting Field of a Polynomial, *ANTS 2006*, Springer LNCS 4076, 124–140.

Fachbereich Mathematik, Universität Kaiserslautern, Postfach 3049, 67653 Kaiserslautern
E-mail address: `fieker@mathematik.uni-kl.de`

Mathematisches Institut der Universität Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
E-mail address: `klueners@math.uni-paderborn.de`